Detection of Ridges and Ravines using Fuzzy Logic Operations

Kyoung-Min Kim, Joong-Jo Park

요 약

영상처리에 의한 물체 해석에 있어서 선의 검출은 중요한 역할을 하는데, 영상에서 선은 ridge과 ravine을 검출함으로서 얻을 수 있다. 본 논문에서는 local min 및 local max 연산을 사용하여 ridge와 ravine을 검출하는 기법을 제시한다. 본 기법은 이들 연산의 침식 및 팽창 특성을 이용하여 방향 정보를 구함이 없이 ridge와 ravine을 검출할 수 있으며, 기존의 해석적 방법에 비해 매우 단순하고 효과적인 방법이다. 실험을 통해 본 기법이 효능을 보인다.

Abstract

In object analysis, line and curve finding plays a universal role. And, it can be accomplished by detecting ridges and ravines in digital gray-scale images. In this paper, we present a new method of detecting ridges and ravines by using local min and max operations. This method uses erosion and dilation properties of these fuzzy logic operations and requires no information of ridge or ravine direction, so that the method is simple and easy in comparison with the conventional analytical methods. The experimental results show that the technique has a strong ability in finding ridges and ravines.

1. Introduction

In object analysis, line and curve finding plays a universal role, and lines and curves in the gray scale image can be obtained through detection of ridges or ravines. Therefore, one important part of the computer vision algorithm lies in the detection of ridge and ravine pixels [1,4-7]. A digital ridge (ravine) occurs on a digital image when there is a simply connected sequence of pixels with gray level intensity values that are significantly higher (lower) in the sequence than those neighboring the

*Department of Electrical Engineering Yosu National University
**Department of Control and Instrumentation Engineering Gyeongsang National University
접수일자 : 2000. 10. 6.

sequence. A ridge (ravine) occurs where there is a local maximum (minimum) in the direction across the ridge (ravine) line. To determine ridges and ravines analytically

[1,6,7], we need to use the neighborhood of a pixel and compute directional derivatives, ie., the direction across the ridge or ravine line is obtained by computing the extremum of the second directional derivatives, and the ridge peak or the ravine bottom is determined by checking the values of the first directional derivatives in the same direction. However this method has some practical complications when applied to the digital image.

This paper presents a simple method for detecting ridges and ravines in the digital image by using Fuzzy logic operations (local min and max operations). Nakagawa and Rosenfeld [8], inspired by Fuzzy set theory, suggested the operations of local min and max. These operations behave analogously to erosion and dilation on gray scale images, which are the basis for most of the gray scale morphological operations. Therefore, the combinations of the local min and max operations can be used to approximate various gray scale morphological operations. Of the morphological operations, opening generally breaks narrow isthmuses and eliminates thin protrusions. Closing, conversely, fuses long thin gulfs and eliminates small holes. These properties can be used in detecting ridges and ravines in gray scale images. In this paper, opening and closing are implemented by using the combinations of local min and max operations and used for detecting ridges and ravines in gray scale images. This method needs no information on direction of the ridge or ravine line, and can detect ridges and ravines of various pixel width simply.

2. Definition of Image by Fuzzy set theory

Since a gray scale image possesses some ambiguity within the pixels due to the possible multi-valued levels of brightness, it is justified to apply the concept and logic of fuzzy set rather than ordinary set theory to an image processing problem [10].

Let $\Pi_\phi = \{\mu_\phi^{ij}, i=1,2\ldots,I; j=1,2\ldots,J\}$ be the fuzzy set representation of the pattern corresponding to an IxJ, L-level image array, where each element μ_ϕ^{ij} is a fuzzy singleton that represents the degree of membership of spatial coordinate (i,j) to the fuzzy subset given by ϕ. The various sets ϕ used in image processing generally involve notions such as gray level, color, texture and shape. In this paper the brightness property of the image is used, therefore the given image is represented as an array of IxJ fuzzy *singletons, each with a membership value equal to* the normalized gray level of the image at that point.

3. Local min and max operations

Shrinking and *expanding* operations on two-valued image are useful for noise removal and segmentation. In a two-valued image consisting of 0's and 1's, shrinking the 1's is equivalent to computing the logical AND of each pixel with its neighbors, and expanding the 1's is equivalent to logically ORing each pixel with its neighbors. One limitation of these is that they can only be applied to binary images. Therefore all real (gray scale) images first have to be thresholded before they can be accessed by them. In trying to cope with the multi-valued membership function, fuzzy logic has extended the Boolean concepts of AND and OR to min and max respectively. For more general operations which behave the same way as the shrink/expand pair do, Nakagawa and Rosenfeld [8] suggested local min and local max operations. The local min operation replaces the membership value at each spatial location (i,j) by the minimum membership value of itself and all immediate neighboring pixels, and the local max operation replaces the membership value at (i,j) by the maximum value of itself and its neighbors. These operations can be used to erode and dilate the multi-valued image.

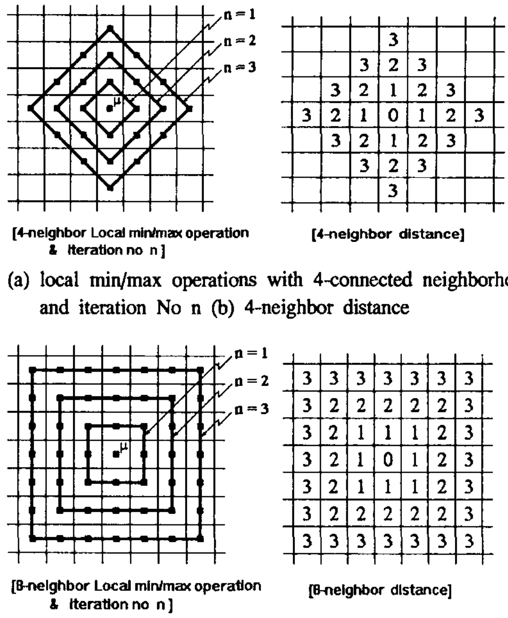
Mathematically, erosion operation for a pixel at (i,j) is written as

$$EROSON\{\mu_\phi^{ij}\} = \min_R\{\mu_\phi^{ij}\} \quad \cdots\cdots (1)$$

where, $\min_R$ is the notation for the local min operator over the region R. The dilation operation is mathematically given by

$$DILATION\{\mu_\phi^{ij}\} = \max_R\{\mu_\phi^{ij}\} \quad \cdots\cdots (2)$$

where, $\max_R$ is the notation for the local max operator over the region ***R***. Local min and max operations belong to the neighborhood operation. In the neighborhood operation, the output at a given pixel position is determined by the values of the input pixels in some neighborhood around the given input position. A neighborhood might be small or large. If we use a symmetric neighborhood centered around the given position, the neighborhood operation with a large neighborhood could be achieved by performing the neighborhood operation with a small neighborhood recursively. Figure 1 shows the range of neighborhood which affects the given center pixel when the local min or max operation is performed recursively. Figure 1(a) is the case where a 4-connected neighborhood is used as the region ***R*** in $\min_R$ or $\max_R$ operator. If these operators are applied ***n*** times to the image, the resulting range of neighborhood which affects the given center pixel is the area composed of the pixels within the 4-neighbor distance equal to n shown in figure 1(b). Similar comments hold for 8-connected neighborhood as shown in figure 1(c) and 1(d). Therefore, instead of using the different sizes of neighborhood, by determining the local min and max operations with some minimal size of neighborhood and applying these operations recursively to the image, the required erosion and dilation could be achieved.



(a) local min/max operations with 4-connected neighborhood and iteration No n (b) 4-neighbor distance

(c) local min/max operations with 8-connected neighborhood and iteration No n (d) 8-neighbor distance

Figure 1. Recursive applications of local min/max operations and its influenced range

4. Detection of ridges and ravines

The local min and max operations behave analogoulsy to erosion and dilation on gray scale images [8-10]. The erosion and dilation are the bases for most of the morphological operations [1-3]. Therefore, we can extend the application areas of local min and max operations by refering the properties of several basic gray scale morphological operations.

In gray scale morphological methods, the gray scale dilation of image ***f*** by image (structuring element) ***b***, denoted $f \oplus b$, is defined as

$$(f \oplus b)(s,t) = \max\{f(s-x, t-y) + b(x,y) \mid (s-x, t-y) \in D_f;\ (x,y) \in D_b\} \quad \cdots\cdots (3)$$

And, the gray scale erosion of f by b, denoted f(b, is defined as

$$(f \ominus b)(s,t) = \min\{f(s+x, t+y) \quad b(x,y) \mid (s+x, t+y) \in D_f; \; (x,y) \in D_b\} \quad \cdots\cdots (4)$$

where D_f and D_b are the domains of f and b, respectively.

Opening and closing are two other important morphological operations for this process. The gray scale opening of image f by b, defined $f \circ b$ is defined as

$$f \circ b = (f \ominus b) \oplus b \quad \cdots\cdots (5)$$

And the gray scale closing of f by b, denoted $f \bullet b$, is defined as

$$f \bullet b = (f \oplus b) \ominus b \quad \cdots\cdots (6)$$

Generally, the effects of the gray scale opening are to remove small bright details from an image, while leaving the overall gray levels and larger bright features relatively undisturbed. The gray scale closing tends to remove dark details from an image, while leaving bright features relatively undisturbed. From these properties, the gray scale opening and closing can be used to extract image components such as ridges and ravines.

These properties of gray scale opening and closing can be obtained by using the combinations of the local min and max operations.

For an image

$$\Pi_G = \{\mu_G^{ij}, i = 1,2,K, I, j = 1,2,K, J\}$$

where μ_G^{ij} is a membership function, which represents the brightness property of the image, the effect of gray scale opening can be obtained by

$$OPENING\{\mu_G^{ij}\} = \max_R\{\min_R\{\mu_G^{ij}\}\}; \quad \mu_G^{ij} \in \Pi_G \quad \cdots\cdots (7)$$

And the effect of gray scale closing can be obtained by

$$CLOSING\{\mu_G^{ij}\} = \min_R\{\max_R\{\mu_G^{ij}\}\}; \quad \mu_G^{ij} \in \Pi_G \quad \cdots\cdots (8)$$

where, the region R in $\min_R$ and $\max_R$ operators corresponds to the structuring element in the morphology. Then, the positions of ridge pixels in the image can be detected by computing the difference of the original image Π_G and its opened version, i.e., the ridge image is obtained by

$$\Pi_{RIDGE} = \{\mu_{RIDGE}^{ij}; i = 1,2,K, I, j = 1,2,K, J\}, \quad 0 \le \mu_{RIDGE}^{ij} \le 1 \quad \cdots\cdots (9)$$

where

$$\mu_{RIDGE}^{ij} = \begin{cases} 1 & if \ \mu_G^{ij} - \max_R\{\min_R\{\mu_G^{ij}\}\} > 0, \ \mu_G^{ij} \in \Pi_G \\ 0 & otherwise \end{cases}$$

In Eq.(9), if we use a 4- or 8-connected neighborhood as the region R in $\min_R$ and $\max_R$ operators, ridges not exceeding 2-pixel width can be detected. Wider ridges can be detected by extending Eq.(9) as follows:

$$\Pi_{RIDGE\_n} = \{\mu_{RIDGE\_n}^{ij}; i = 1,2,K, I, j = 1,2,K, J\}, \quad 0 \le \mu_{RIDGE\_n}^{ij} \le 1 \quad \cdots (10)$$

where

$$\mu_{RIDGE\_n}^{ij} = \begin{cases} 1 & if \ \mu_G^{ij} - \max_R^{(n)}\{\min_R^{(n)}\{\mu_G^{ij}\}\} > 0, \ \mu_G^{ij} \in \Pi_G \\ 0 & otherwise \end{cases}$$

where (n) is the number of iterations of $\min_R$ and $\max_R$ applied to the image.

In Eq.(10), if we use a 4- or 8-connected neighborhood as the region R in minR and maxR operators, ridges not exceeding (2xn)-pixel width can be detected. If we need not only the positions but also the height information of ridges, we can obtain those by modifying Eq.(10) as

$$\Pi_{RIDGE\_n} = \{\mu_{RIDGE\_n}{}^{ij}; i = 1,2,K\ ,I, j = 1,2,K\ ,J\},\quad 0 \le \mu_{RIDGE\_n}{}^{ij} \le 1 \qquad \cdot\cdot (11)$$

where

$$\mu_{RIDGE\_n}{}^{ij} = \begin{cases} \mu_G{}^{ij} & \text{if } \mu_G{}^{ij} - \max_R{}^{(n)}\{\min_R{}^{(n)}\{\mu_G{}^{ij}\}\} > 0,\ \mu_G{}^{ij} \in \Pi_G \\ 0 & \text{otherwise} \end{cases}$$

Similarly, the position of ravine pixels in the gray scale image can be detected by computing the dif ference of the original image and its closed version, i.e., the ravine image Π_{RAVINE} is obtained by

$$\Pi_{RAVINE\_n} = \{\mu_{RAVINE\_n}{}^{ij}; i = 1,2,K\ ,I, j = 1,2,K\ ,J\},\quad 0 \le \mu_{RAVINE\_n}{}^{ij} \le 1 \qquad \cdot\cdot (12)$$

where

$$\mu_{RAVINE\_n}{}^{ij} = \begin{cases} 1 & \text{if } \mu_G{}^{ij} - \max_R{}^{(n)}\{\min_R{}^{(n)}\{\mu_G{}^{ij}\}\} > 0,\ \mu_G{}^{ij} \in \Pi_G \\ 0 & \text{otherwise} \end{cases}$$

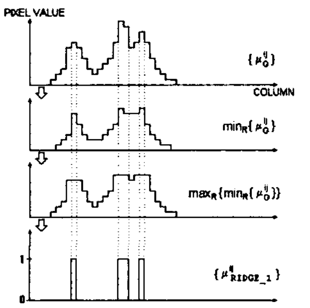
In Eq.(12), if we use a 4- or 8-connected neighborhood as the region R in $\min_R$ and $\max_R$ operators, ravines not exceeding (2xn)-pixel width can be detected. If we need not only the positions but also the height information of ravines, we can obtain those by modifying Eq.(12) as

$$\Pi_{RAVINE\_n} = \{\mu_{RAVINE\_n}{}^{ij}; i = 1,2,K\ ,I, j = 1,2,K\ ,J\},\quad 0 \le \mu_{RAVINE\_n}{}^{ij} \le 1 \qquad \cdots (13)$$

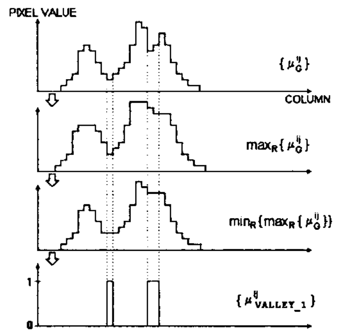
where

$$\mu_{RAVINE\_n}{}^{ij} = \begin{cases} \mu_G{}^{ij} & \text{if } \mu_G{}^{ij} - \max_R{}^{(n)}\{\min_R{}^{(n)}\{\mu_G{}^{ij}\}\} > 0,\ \mu_G{}^{ij} \in \Pi_G \\ 0 & \text{otherwise} \end{cases}$$

In this case, zero valued ravine pixels in the original image are removed, therefore some labeling on those are required. Figures 2 shows the procedures detecting ridges and ravines in a scan line of a gray scale image by using Eqs.(11) and (13) with n=1, respectively.



(a) ridge detection



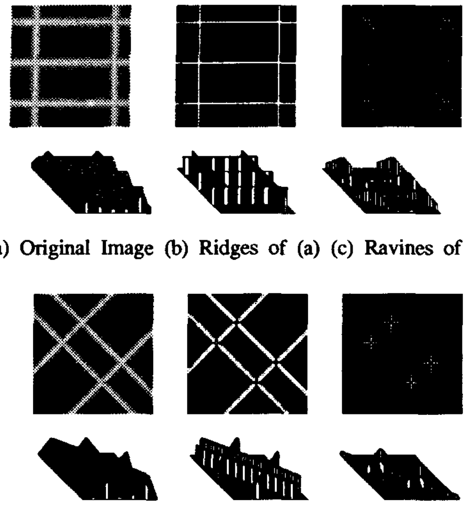
(b) ravine detection

Figure 2. The procedures of detecting ridges and ravines

5. Experimental Results

To illustrate the power of the ridge-ravine detection scheme, we show results on pattern images and real images. Figure 3 shows the results on pattern images and their 3-D view. Figure 3(a) is a input pattern image and figure 3(b) and 3(c) show the detected ridges and ravines, respectively. Figure 3(d) is another pattern image and figure 3(e) and 3(f) show the detected ridges and ravines, respectively. To obtain these results, Eqs.(11) and (13) with n=1 are used.

The real image of the airphoto scene is shown in figure 4(a). Figure 4(b) shows the ridges alone obtained by using Eq.(11) with n=1 and thresholding appropriately, and figure 4(c) shows the ridges overlayed with the airphoto scene.



(a) Original Image (b) Ridges of (a) (c) Ravines of (a)

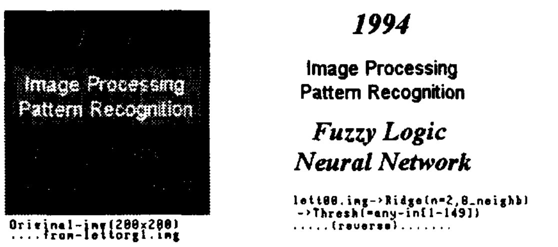
(d) Original Image (e) Ridges of (d) (f) Ravines of (d)

Figure 3. The computed ridges and ravines for the pattern images

(a) Original Image (b) Ridges of (a) (c) Ridges overlayed with original image

Figure 4. The computed ridges for the airphoto scene

Figure 5(a) is the image composed of different characters on different backgrounds in gray levels. We can segment characters of different width from the background by using our ridge detection method. Figure 5(b) shows the characters obtained by using Eq.(10) with n=2.



(a) Original image (b) Ridges of (a)

Figure 5. The computed ridges for the character image

6. Conclusion

We have presented a method of detecting ridges and ravines in gray scale images by using local min and max operations. This method stems from the idea that, of the gray scale morphological operations, the opening operation tends to remove small bright details and the closing operation removes dark details from the gray scale image. In this method, the properties of opening and closing are implemented by using the combinations of fuzzy logic operations (local min and max operations). This method needs no information of ridge or ravine direction, and can detect ridges and ravines of various pixel width simply.

References

[1] Haralick and Shapiro, Computer and Robot Vision, vol. 2, Addison-Wesley publishing co., 1992

[2] R.C.Gonzalez and R.E.Woods, Digital Image Processing, Addison-Wesley publishing co., 1992

[3] C.R.Giardina and E.R.Dougherty, Morphological Methods in Image and Signal Processing, Prentice-Hall, 1988

[4] T.K.Peucker and D.H.Douglas, "Detection of surface-specific points by local parallel processing of discrete terrain elevation data", Computer Graphics and Image Processing 4, pp.375-387, 1975

[5] E.G.Johnston and A.Rosenfeld, "Digital detection of pits, peaks, ridges and ravines", IEEE Trans. on Syst.,Man, and Cybern.,pp.472-480, July, 1975

[6] R.M.Haralick and L.Watson, "A facet model for image data", Computer Graphics and Image Processing 15, pp.113-129, 1981

[7] R.M.Haralick, "Ridges and valleys on digital images", Computer Vision, Graphics, and Image Processing 22, pp.28-38, 1983

[8] Y. Nakagawa and A. Rosenfeld, "A Note on the Use of Local min and max Operations in Digital Picture Processing", IEEE Trans. on Syst., Man, and Cybern., vol. SMC-8, No.8, pp. 632-635, 1978.

[9] V. Goetcherian, "From Binary to Grey Tone Image Processing Using Fuzzy Logic Concepts", Pattern Recognition, Vol.12, pp.7-15, 1980.

[10] M. M. Gupta and G. K. Knopf, "Fuzzy Logic in Vision Perception", SPIE Intelligent Robots and Computer Vision XI, Vol.1826, pp.300-316, 1992.

Kyoung-Min Kim was born in Korea in 1966. He received the B.S., M.S. and Ph. D. degrees from the Kore University in 1988, 1991 and 1996 respectively. Since 1997 he is currently assistant professor at Yosu National University, Korea. His current interests include the area of computer vision, pattern recognition , fuzzy and neural network..

Joong-Jo Park received the B.S., M.S. and Ph. D. degrees from the Kore University in 1981, 1983 and 1995 respectively. Since 1997 he is currently associate professor at Gyeongsang National University, Korea. His current interests include the area of computer vision, pattern recognition , fuzzy and neural network.

김 경 민(Kyoung-Min Kim)
1988년 고려대 전기공학과 졸업.
동대학원 석사(1991), 박사(1996).
1997년~현재 여수대학교 전기공학과 조교수
관심분야 : 신호처리 및 컴퓨터 비젼, 퍼지 및 신경회로망 응용.

박 중 조(Joong-Jo Park)
1981, 1983, 1995 고려대 전기공학과 학사 석사 박사.
1996년~현재 경상대학교 전기전자공학부 부교수.
관심분야 : 컴퓨터비젼, 패턴인식, 퍼지 및 신경회로망.